Práctica: Autenticación y autorización de usuarios con JAAS

Objetivo: Implementar un Sistema de Autenticación y autorización de forma segura con los mecanismos que proporciona J2EE.

Desarrollo:

1. Crea un nuevo proyecto Web dinamico llamado JAAS, y genera la siguiente estructura de ficheros:

   Webcontent
   - privado
     - administrador
       - admin.jsp
     - bienvenido.jsp
   - index.jsp
   - login.jsp
   - loginError.jsp

   1.1. Index.jsp es la página de bienvenida, únicamente trata de dirigir al usuario a la parte privada

```
<%response.sendRedirect("privado/Bienvenido.jsp");%>
```

   1.2. La página login.jsp tiene el form para realizar el login, observar que el action y el nombre los campos están predefinidos a la hora de utilizar el mecanismo de autenticación de J2EE.

```
<form action="j_security_check" name="j_security_check" method="post">
	<table border="0">
		<tr><td>Usuario:</td><td><input type="text" name="j_username" /></td></tr>
		<tr><td>Password:</td><td><input type="password" name="j_password" /></td></tr>
		<tr><td colspan="2"><input type="submit" value="Login"></input></td></tr>
	</table>
</form>
```

   1.3. La página loginError.jsp muestra simplemente un mensaje de error:

```
h3>Credenciales de acceso no validas</h3>
<a href="#" onclick="javascript:history.back()">Volver</a>
```

   1.4. Bienvenido.jsp muestra el nombre del usuario que ha iniciado sesión

```
<h1>Bienvenido al Sistema <%=request.getUserPrincipal().getName() %></h1>
```

   1.5. Admin.jsp, muestra un mensaje para el administrador

2. Creamos una nueva política de seguridad en el fichero login-config.xml situado en "C:\jboss-5.1.0.GA\server\default\conf" donde indicamos que utilizamos el proveedor de usuarios y claves almacenadas en ficheros de propiedades:

```
<application-policy name="MiPolitica">
	<authentication>
		<login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"
			flag="required">
			<module-option name="usersProperties">jaas/jaas-users.properties
			</module-option>
			<module-option name="rolesProperties">jaas/jaas-roles.properties
			</module-option>
		</login-module>
	</authentication>
</application-policy>
```

2.1. Crear los correspondientes ficheros de propiedades:

2.1.1. Jaas\jaas-roles.properties

```
alex=Usuario
pepe=Administrador
```

2.1.2. Jaas\Jaas-users.properties

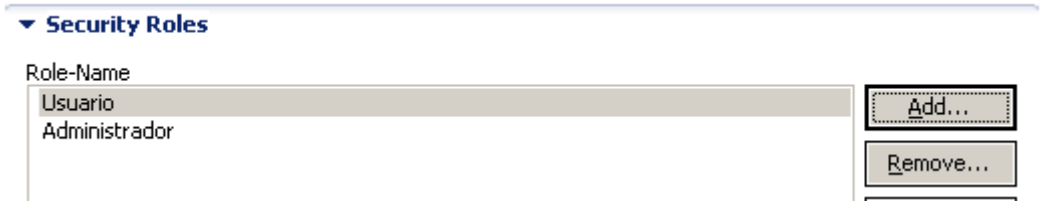
```
alex=123abc.
pepe=admin
```

2.2. Crear en la carpeta WEB-INF el fichero jboss-web.xml donde se indica el proveedor que se va a utilizar:

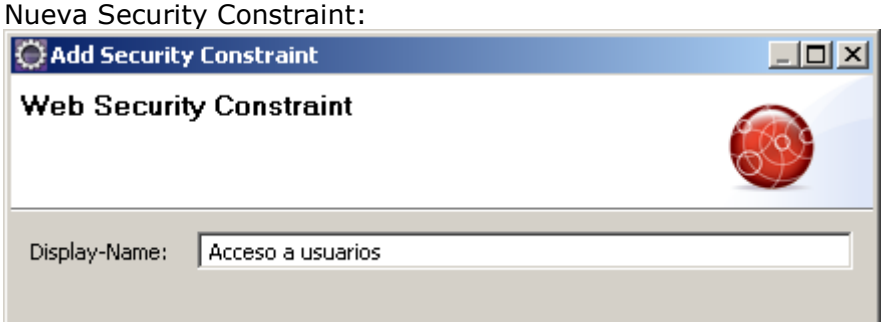
```
<jboss-web>
<security-domain>java:/jaas/MiPolitica</security-domain>
</jboss-web>
```

3. Crear los roles y políticas de seguridad para controlar que roles pueden acceder a las diferentes partes de la aplicación, a través del fichero web.xml de la aplicación:

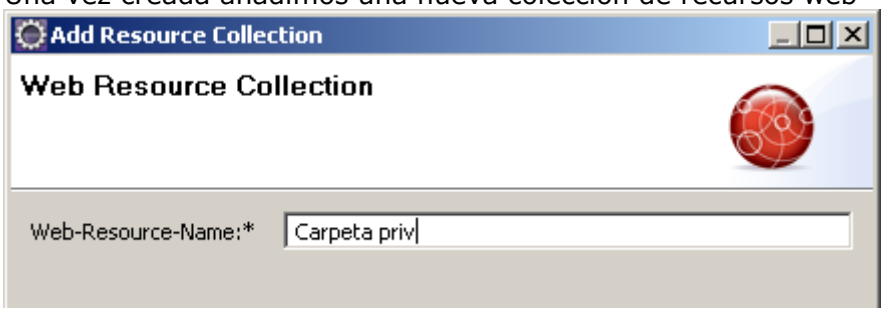
3.1. Añade el role Usuario y el role Administrador



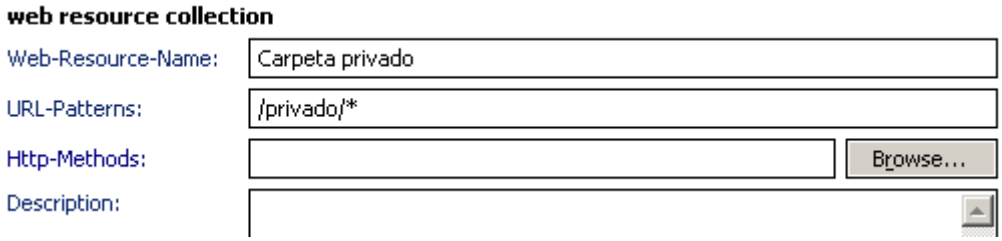
3.2. Nueva Security Constraint:



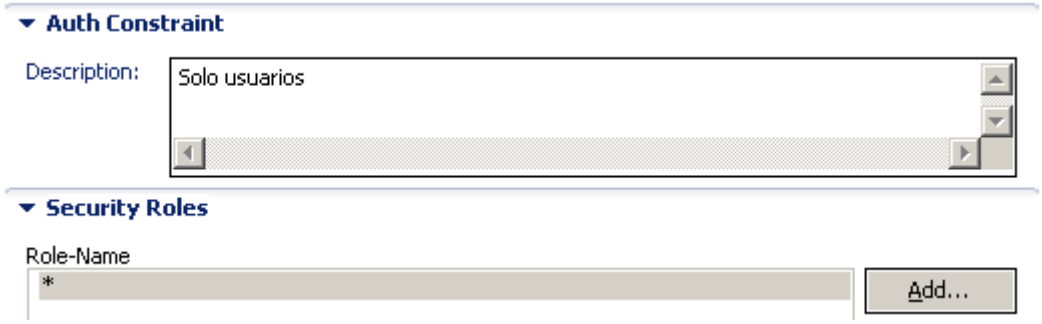
3.3. Una vez creada añadimos una nueva colección de recursos web



3.4. Indicamos la URL del recurso recién creado



3.5. Se añade una restricción marcando todos los roles



3.6. Repetir el proceso para proteger la carpeta administrador solo a los usuarios con Role administrador

4. Prueba la aplicación, comprobando que solo los usuarios con los roles adecuados pueden acceder a los recursos adecuados